

PERFORMANCE MONITORING AND OPTIMIZATION

**After reading this chapter and completing the exercises,
you will be able to:**

- ◆ Effectively use performance monitoring tools
- ◆ Establish a baseline
- ◆ Recognize acceptable and unacceptable performance thresholds
- ◆ Provide solutions to performance bottlenecks

Monitoring server performance is a critical function in the enterprise network. Effectively monitoring a server is a scientific process that separates perceived server “slowdowns” from true performance degradation based on empirical data. The data you collect is also useful when proposing major changes to the enterprise network architecture or new equipment. Using the proper tools will help you to obtain accurate data. Each respective network operating system (NOS) discussed in this book has its own set of monitoring and performance tools, and you should know what those tools are and how to use them.

Whichever tool applies to your operating system, it is critical to establish a baseline. The baseline is the fulcrum that balances subjective perceptions of network performance against objective data, and establishes what is to be considered acceptable performance. For systems that do not perform within acceptable parameters, you must accurately determine which server or network components are causing the bottleneck, and take appropriate action.

MONITORING THE SERVER

It's Monday morning. You sit down at your desk and the telephone rings with a complaint that logon is slow. You attempt to log on to your own computer but before you can type a password, another call comes. Same problem. The moment you hang up, the telephone rings again and two co-workers are at your cube wanting to know why they can't log on. Your best response at this point is, "I'm working on it."

So where do you start? Luckily, you have established performance baselines for the logon server. Therefore, you begin by checking the current performance of the server against the baseline. Your Windows 2000 Performance Monitor tells you that the amount of memory in use is extremely high compared with the baseline. You identify the process consuming the memory and discover that the new monitoring agents installed on this server are using more resources than predicted. You terminate the processes temporarily until more memory can be added to the server. Within 10 minutes, the problem is solved and users are logging on quickly.

Performance monitoring—observing, measuring, and recording the performance of critical server and network resources—is essential for troubleshooting and maintaining a network. There are several reasons to monitor servers:

- To become familiar with your server's "normal" performance so you know when there is a problem
- To notice impending problems and prevent them before they occur
- To pinpoint existing problems and identify solutions
- To aid in resource and capacity planning

Performance monitoring is the best tool for systematic troubleshooting, capacity planning, and checking on the "health" of servers. It can mean the difference between being unprepared when a problem comes up, or anticipating a problem and correcting it before users even notice.

Table 11-1 shows some typical server areas that can be monitored.

Table 11-1 Server Monitoring Activities

Monitor ...	To Determine
CPU	CPU utilization and performance
RAM	Memory shortage or damaged memory
Hard disk	Disk performance, capacity, and errors
Paging	Page file size and performance
Caching	Cache allocation and performance
Process	Hung or stopped service or process using high CPU resources
Users	Number of users logged on and types of resources they are accessing

The operating systems discussed in this book use different tools to monitor the performance of server components listed in Table 11-1, but they are used in similar ways: to establish performance baselines, to measure current performance (and perhaps compare it to a baseline), and to keep logs of performance over time.



There are many, many possible performance measures and results, and performance monitoring can, at first glance, be a little overwhelming. The key is to focus on the most significant resources. With performance monitoring, “less is more.” Focusing your attention on the most critical resources will help to achieve the most effective results.

USING MONITORING TOOLS

The performance monitoring concepts presented in the previous section are consistent across all platforms. The specific tools and functions vary according to operating system. We will discuss tools for:

- IBM OS/2
- Linux
- NetWare
- Windows NT 4.0
- Windows 2000



The main focus of performance monitoring should be the accurate gathering and interpretation of performance data regardless of which operating system or tool you use. This chapter provides more in-depth coverage of the monitoring tools for Windows NT 4.0 and Windows 2000 as examples, rather than detailing all monitoring functions for all of the operating systems.

Among all the NOSs, there are literally hundreds of different measures of performance. Although it is not practical to define each performance measure, you should be aware of the main tools and resource categories for each operating system. After a brief tour of the primary performance monitoring tools in several operating systems, you’ll learn about establishing a baseline and how to use monitoring results for troubleshooting and planning. This chapter also helps you to identify major performance bottlenecks and propose solutions for each area.

IBM OS/2

System Performance Monitor/2 (SPM/2) is designed to analyze hardware and software in the OS/2 environment. SPM/2's primary features for monitoring critical resources are:

- *SPM/2 Monitor*—A Presentation Manager application that displays performance data in graph form. Data is summarized from real-time input.
- *Data Collection Facility*—A tool that gathers data for system resources in use. The information can be displayed with SPM/2 Monitor in the Presentation Manager window as graphic or real-time output. The data can also be logged to a file using the Logging Facility.
- *Report Facility*—A program that generates reports from collected data and is far more detailed than SPM/2 Monitor. Data collected into the Report Facility can be displayed or exported in three formats: summary, tabular, or dump.
- *Logging Facility*—A tool that accesses data from the Data Collection Facility and saves it to log files. Log files are available to the Report Facility.

SPM/2 uses a distributed management approach to performance monitoring. The SPM/2 application is installed on a monitoring station. Servers designated for monitoring collect data and distribute it back to the monitoring station for analysis as shown in Figure 11-1. The advantage of this approach, not only for OS/2 but for any NOS that supports it, is that the overhead required to run the monitoring software does not skew the monitoring results.

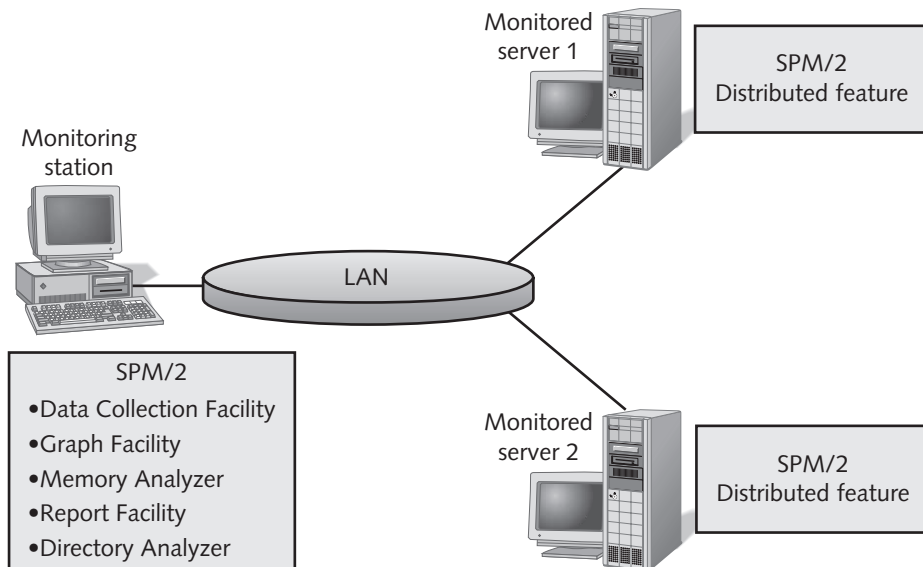


Figure 11-1 SPM/2 uses a distributed feature to remotely monitor servers

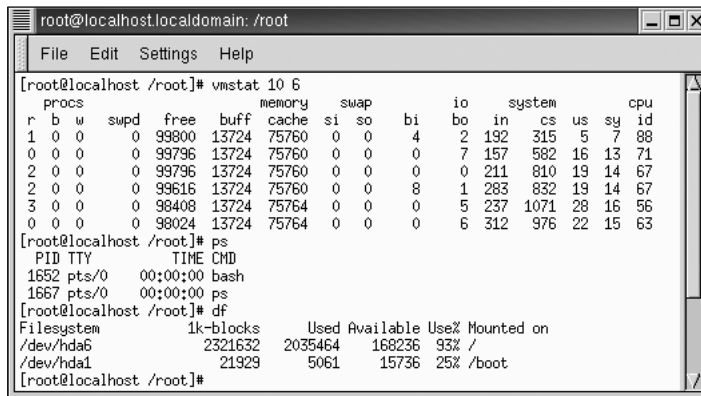
Linux

UNIX tools are used to monitor performance on Linux systems. These tools are command-line utilities that provide statistics on CPU usage, memory, disk I/O, and network connections. Although there are multiple UNIX/Linux utilities available for specific monitoring purposes, some of the most commonly used are shown in Table 11-2.

Table 11-2 Common Linux/UNIX Performance Tools

Command	Function
<i>vmstat</i>	Provides information on memory usage, CPU, and interrupts
<i>ps</i>	Lists all processes current running on the system
<i>df</i>	Lists disk space used and available
<i>top</i>	Shows top several processes running and amount of resources they consume

Figure 11-2 shows the output from the *vmstat*, *ps*, and *df* commands.



```

root@localhost.localdomain: /root
File Edit Settings Help
[root@localhost /root]# vmstat 10 6
procs          memory          swap          io          system          cpu
 r  b  w    swpd    free    buff    cache    si    so    bi    bo    in    cs    us    sy    id
1  0  0      0  99800  13724  75760    0    0    4    2  192   315    5    7   88
0  0  0      0  99796  13724  75760    0    0    0    7  157   582   16   13   71
2  0  0      0  99796  13724  75760    0    0    0    0  211   810   19   14   67
2  0  0      0  99616  13724  75760    0    0    8    1  283   832   19   14   67
3  0  0      0  98408  13724  75764    0    0    0    5  237  1071   28   16   56
0  0  0      0  98024  13724  75764    0    0    0    6  312   976   22   15   63
[root@localhost /root]# ps
PID TTY          TIME CMD
1652 pts/0      00:00:00 bash
1667 pts/0      00:00:00 ps
[root@localhost /root]# df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/hda6         2321632    2035464    168236   93% /
/dev/hda1         21929      5061     15736   25% /boot
[root@localhost /root]#

```

Figure 11-2 UNIX utilities *vmstat*, *ps*, and *df* provide a snapshot of current system activity

The *vmstat* tool provides real-time performance statistics for several resources. For example, when system performance slows, you can use *vmstat* to provide a quick snapshot of CPU load average to determine what process is causing a bottleneck. The syntax for the utility looks like this:

```
vmstat seconds #OfReports
```

If you wanted to take a snapshot every 10 seconds and create a total of six reports, you would type *vmstat 10 6*, which is what Figure 11-2 shows. If you do not specify the number of reports, the utility runs continuously until you issue the Ctrl+C command.

In addition to displaying the top consuming resources, the *top* command also provides other information such as the number of users logged on, the amount of memory consumed, and how much is swapped out to the swap file. Figure 11-3 shows sample output of the *top* command.

```

root@localhost.localdomain: /root
File Edit Settings Help

      4      5      , load average: 0.16, 0.18, 0.15
9 ccesses: 92 sleeping, 2 running, 0 zombie, 0 stopped
C ates: 7.4% user, 13.0% system, 0.0% nice, 79.4% idle
M 257596K av, 179984K used, 77612K free, 185008K shrd, 11156K buff
S 105800K av,      0K used, 105800K free

PID USER  PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM TIME COMMAND
1201 root    11   0 4692 4692 3320 S   9.8  1.8 1:30 gtop
1024 root     3   0 26596 25M 2796 R   3.7 10.3 0:57 %
1316 root     1   0 4792 4792 3048 S   2.3  1.8 0:02 ksnapshot
1195 root     3   0 3684 3684 2892 S   1.9  1.4 0:36 multiloop_apple
1336 root     3   0 1068 1068 816 R   1.3  0.4 0:03 top
1093 root     1   0 4468 4468 2072 S   0.3  1.7 0:14 sawfish
1187 root     0   0 4228 4228 3128 S   0.3  1.6 0:03 tasklist_applet
1189 root     0   0 3840 3840 3024 S   0.3  1.4 0:02 deskguide_apple
   1 root     0   0 532 532 468 S   0.0  0.2 0:06 init
   2 root     0   0 0 0 0 SW   0.0  0.0 0:00 kflushd
   3 root     0   0 0 0 0 SW   0.0  0.0 0:00 kupdate
   4 root     0   0 0 0 0 SW   0.0  0.0 0:00 kpiod
   5 root     0   0 0 0 0 SW   0.0  0.0 0:00 kswapd
   6 root    -20 -20 0 0 0 SWK  0.0  0.0 0:00 mdrecoveryd
  61 root     0   0 0 0 0 SW   0.0  0.0 0:00 khubb
 322 root     0   0 612 612 512 S   0.0  0.2 0:00 syslogd
 332 root     0   0 828 828 464 S   0.0  0.3 0:00 klogd
 347 rpc     0   0 576 576 484 S   0.0  0.2 0:00 portmap
 363 root     0   0 0 0 0 SW   0.0  0.0 0:00 lockd
 364 root     0   0 0 0 0 SW   0.0  0.0 0:00 rpciod
 374 rpcuser  0   0 768 768 656 S   0.0  0.2 0:00 rpc.statd
 389 root     0   0 524 524 460 S   0.0  0.2 0:00 apmd
 440 root     0   0 628 628 528 S   0.0  0.2 0:00 automount

```

Figure 11-3 The UNIX *top* command provides a comprehensive snapshot of ongoing system activity



The *top* command automatically refreshes. This can be advantageous when you are troubleshooting, but keep in mind that the refresh itself consumes resources. Remember the additional load when using the *top* command in heavily loaded systems.

If you also use graphical UNIX/Linux utilities, several other performance monitoring tools might be available to you, including the GNOME System Monitor (Figure 11-4) and the Stripchart Plotter (Figure 11-5), which provides a quick graphical snapshot of processor, swap file, network, and PPP activity.

Third-party tools that provide a graphical interface for monitoring are also useful. For example, Computer Associates' Unicenter TNG, an enterprise management software package, provides a graphical interface to monitor performance on Linux as well as most flavors of UNIX.

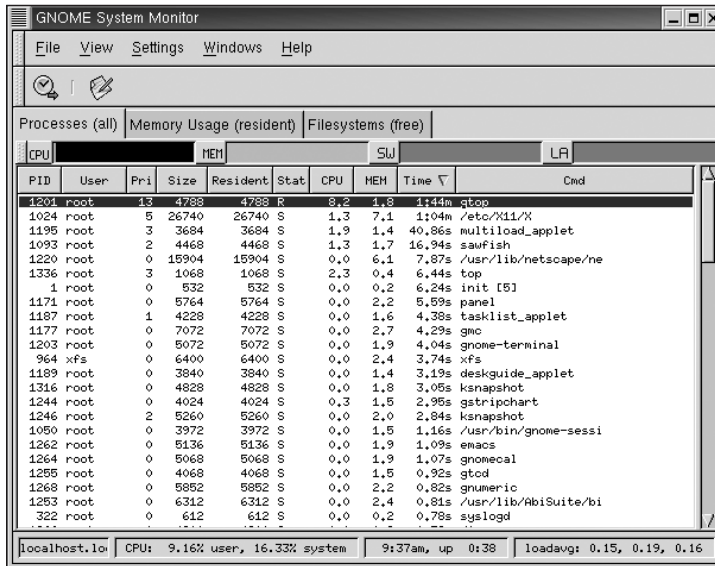


Figure 11-4 The GNOME System Monitor tool

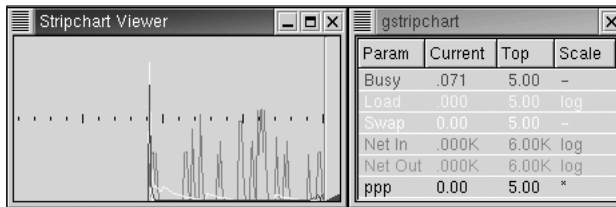


Figure 11-5 The Stripchart Plotter

NetWare

Novell NetWare uses a service known as Traffic Manager to monitor network traffic. Traffic Manager runs on Windows NT computers and uses Windows NT's Performance Monitoring tool to display its data (see next topic).

The Monitor utility is included with NetWare to track server performance (Figure 11-6). Many Novell system administrators will leave this screen on instead of a conventional screen saver! When Monitor is running, four performance indicators are shown:

- *Utilization*: This shows the CPU utilization rate for servicing network requests. If this number is consistently greater than 50–65%, your CPU is a bottleneck. (Specific thresholds are discussed later in this chapter.)
- *Total Cache Buffers*: If this number is quite low, your system will suffer from slow file performance.

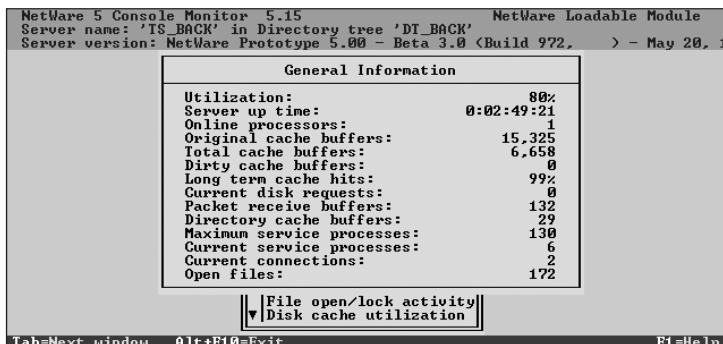


Figure 11-6 The Monitor utility monitors performance under NetWare

- *Current Service Processes*: This indicates outstanding read requests. If a read request is buffered, it means that resources were not available. This may indicate that you need to upgrade your disk controller.
- *Packet Receive Buffers*: This is an indicator that shows packets that are being buffered from workstations.

For a GUI, use the Java-based ConsoleOne.

Administrators running web or FTP services on NetWare servers will probably rely on the Novell Internet Caching System (ICS) utility to track and optimize performance using the ICS caching facility, but it can also be useful for monitoring general server performance and network activity (see Figure 11-7).

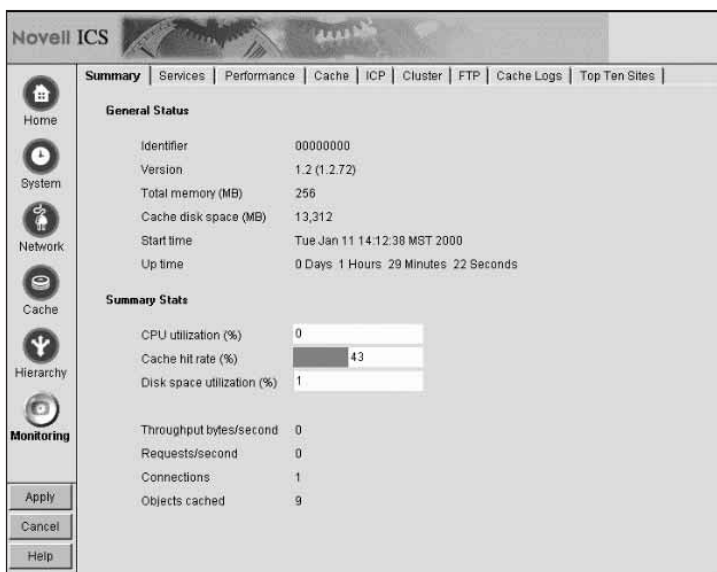


Figure 11-7 The Novell ICS utility

Windows NT 4.0

Performance monitoring on the Windows NT 4.0 operating system uses the Performance Monitor GUI tool. This tool uses objects, instances, and counters to measure performance on local servers or remote systems. Open Performance Monitor on a Windows NT system by clicking Start, pointing to Programs, pointing to Administrative Tools, and then clicking the Performance Monitor icon.



For best results, monitor an NT server from an NT workstation. Running Performance Monitor locally on the server creates an artificial load that can skew performance data. (Remotely monitoring a server can also add to the network load, although the impact is generally minimal.)

ChartView is the default view for Performance Monitor and provides real-time dynamic snapshots of server activity (see Figure 11-8). Snapshots are taken in one-second intervals by default.

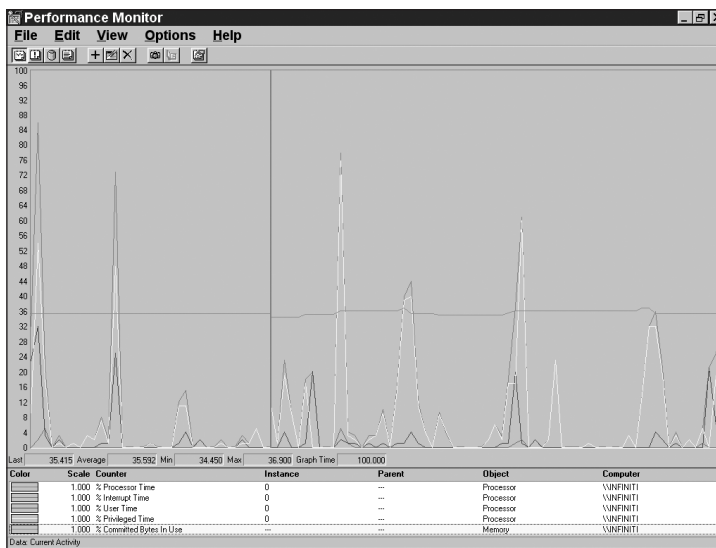


Figure 11-8 Chart View is the default for Windows NT 4.0 Performance Monitor

Performance is measured by choosing objects to monitor. In Performance Monitor, **objects** are resources such as Processor, Memory, PhysicalDisk, and Network Segment. After selecting an object to monitor, you choose specific **counters**, which are measures pertaining to Performance Monitor objects. For example, when monitoring the Processor object, you must specify exactly what it is about the processor you want to monitor by selecting a counter. Some examples of Processor counters are:

- *% Processor Time*—A primary indicator of overall processor activity.

- *Interrupts/Sec*—The average number of hardware interrupts the processor receives and services per second. During an interrupt, normal processes owned by applications, services, and so forth are unable to perform actions, so you want to be sure to watch this counter.
- *% User Time*—The percentage of processor time spent in user mode, which includes applications, environment subsystems, and integral subsystems. Despite the use of the word “user,” this counter is not always tied to user activities per se.
- *% Privileged Time*—The percentage of processor time spent in privileged mode, which is designed for hardware driver activity and operating system components. A high percentage might indicate a failing hardware device or driver that sends out excessive interrupts.

If there is more than one processor on the system, the Processor object will also have multiple **instances** to distinguish one processor from the other. Instances also apply to other resources such as multiple hard disks or multiple NICs. Using instances provides the capability to monitor processors or other components collectively or individually.

You can add objects and counters to the chart that are relevant to the tasks performed by the server. (Information concerning how to determine these objects is presented later in this chapter.)

You can also save settings for objects and counters so that you can return to monitor the same objects and counters at a later date. The simplest way is to save a Performance Monitor file. Once your chart is set, press the F12 key. Enter a name for the file in the Save As dialog box (see Figure 11-9).

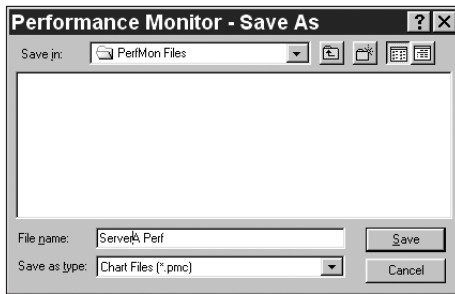


Figure 11-9 Windows NT Performance Monitor settings are saved with a .pmc extension by default

Windows NT 4.0 also uses performance logs to measure historical performance data and to set alerts to call attention to specific threshold breaches. Performance logs are discussed more specifically in the Windows 2000 section.

Windows 2000

Performance monitoring in Windows 2000 uses the Microsoft Management Console (MMC) graphical interface. The Windows 2000 monitoring tool uses objects, instances, and counters in a manner similar to Windows NT 4.0. The steps to open and use the Windows 2000 Performance console, like all management tools in Windows 2000, has changed from NT 4.0. The Performance console can be opened from Administrative Tools or added as a snap-in to an MMC containing other management tools.



A very handy feature of both Windows NT 4.0 and Windows 2000 is the Explain button that appears when you want to add a counter. By clicking it, an explanation appears for the otherwise cryptic counters in the list (see Figure 11-10).

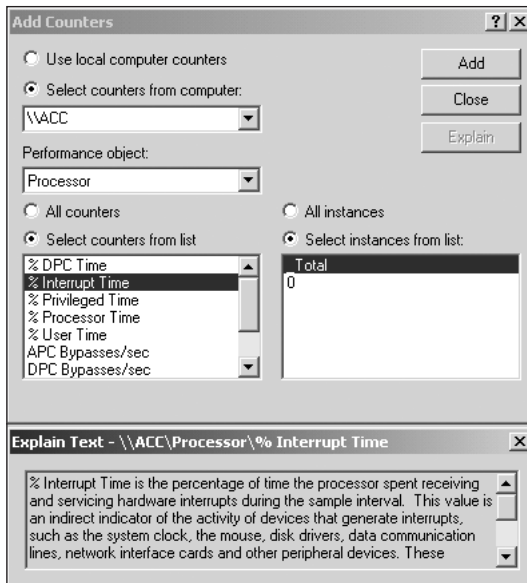


Figure 11-10 The Explain button describes each counter

The Performance console contains two snap-ins: System Monitor, and Performance Logs and Alerts. System Monitor provides real-time snapshots of system resources on local or remote servers (Figure 11-11). The Performance Logs and Alerts snap-in offers two functions:

- Performance logs gather historical performance data over a period of time.
- Performance alerts send messages when designated thresholds are exceeded based on dynamic data.

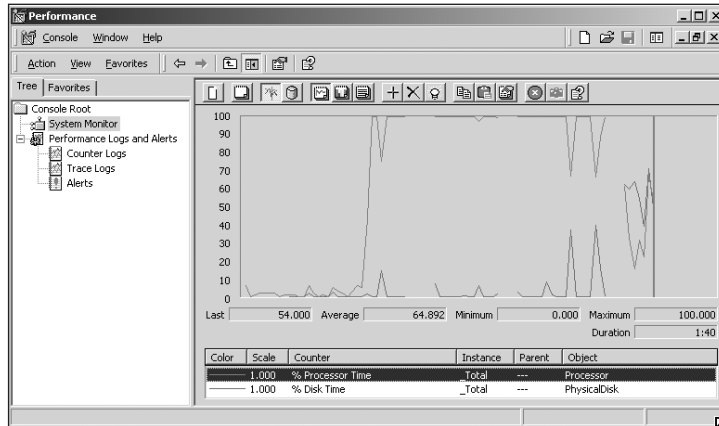


Figure 11-11 The Windows 2000 System Monitor gathering real-time performance data

System Monitor

The Windows 2000 System Monitor relies on objects and counters to display data in the chart. Click the plus sign (+) button in the toolbar above the chart to add objects and counters. From the Add Counters dialog box, you can choose to monitor the local server or a remote server. Figure 11-12 shows the Performance object *PhysicalDisk* selected from the drop-down list. The counter, *% Disk Time*, has also been chosen. By reading the information in the Instances list, we can see that the server named “infiniti” has two physical disks. The disks are labeled 0 and 1. The Instances box provides the capability to monitor the disks individually or collectively. Selecting “_Total” monitors both/all physical disks.

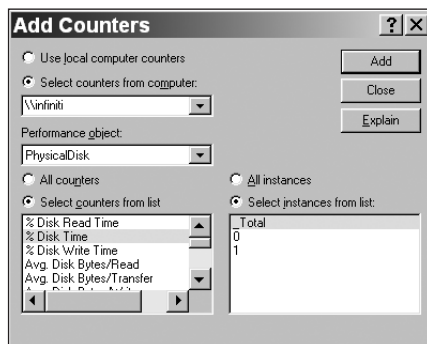


Figure 11-12 The System Monitor Add Counters dialog box

Performance Logs and Alerts

Performance logs monitor resources over a specific period of time. This is termed “historical performance monitoring.” The time can be hours, days, or weeks depending on the situation. The log records the data. When logging is completed, the data can be displayed in a static format in the System Monitor screen.

To begin recording performance logs, you must create a log file:

1. Click the plus sign (+) button to the left of Performance Logs and Alerts.
2. Right-click Counter Logs. Select New Log Settings.
3. Give an intuitive name to the new log. Click OK.

Objects and counters must be added to the log for historical data just as you add objects and counters to System Monitor for real-time data.



The objects and counters are the same in Performance Logs and Alerts as in System Monitor, except that they can be configured to record over a specific period of time and can issue alerts upon reaching a specified threshold.

After choosing the objects and counters and returning to the Counter Log dialog box, there are multiple options for the time and frequency to gather data. Notice that the counter samples data every 15 seconds by default, because it is expected that performance logs will record over a longer duration. You can adjust the data-sampling interval as you like, but if you make it too short, the log files can get quite large and unmanageable. Figure 11-13 displays the General tab of the Counter Log dialog box.

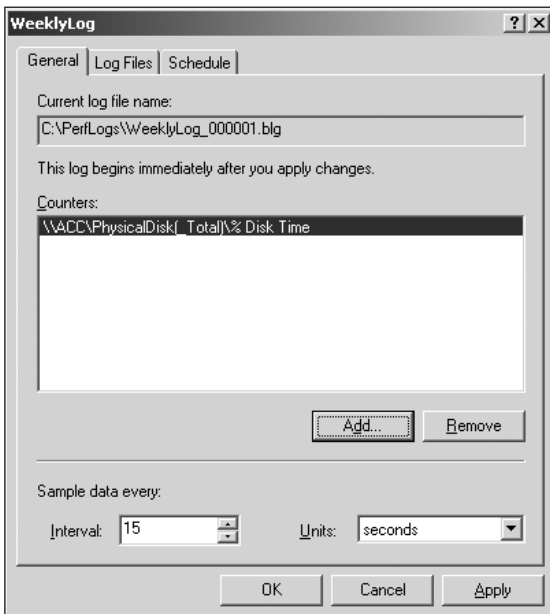


Figure 11-13 The General tab of the Counter Log dialog box

Besides adding counters from the General tab of the interface, you can also select the Log Files tab to specify characteristics of the log file itself, such as maximum size, location, and naming preferences (see Figure 11-14). The Schedule tab shows the total time

frame for the log to record data, as compared to the data-sampling interval shown on the General tab, which determines the frequency of system snapshots.

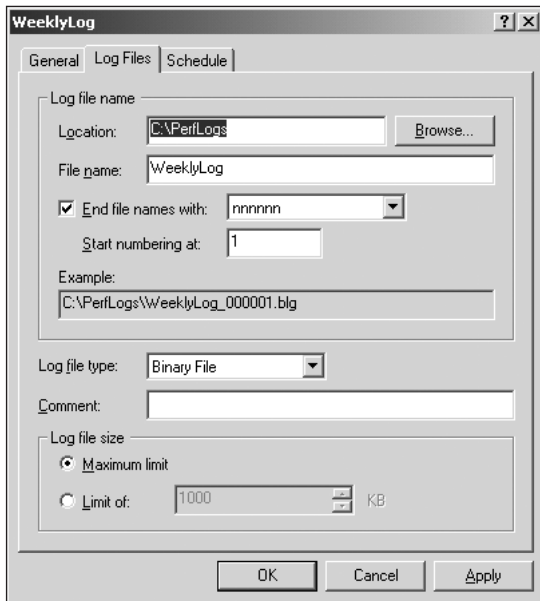


Figure 11-14 The options for the Log Files tab

From the Log Files tab, specify the following items:

- A file location, preferably not on the same disk or system you are monitoring, so as not to skew the results.
- An intuitive file name, so that you and others easily identify the file.
- “End file names with” allows log files to be tagged with sequential numbers or dates.
- The default log file type is binary. Log files can also be saved in formats such as .CSV to enable simple import to databases or spreadsheets.
- Log file size, by default, allows growth potential limited only by the space on the hard disk. This can be limited to a specific size with this option.

Once all parameters are determined, you can start the log manually or schedule it to run automatically in the future. Initiate a manual start from the Performance console as follows:

1. Click the plus sign (+) button on Performance Logs and Alerts.
2. Click Counter Logs. This displays all eligible logs.
3. Right-click the log and select Start.
4. To stop the log manually, right-click the log and select Stop.

The options to schedule the log are displayed in Figure 11-15. Start the log according to time and date. The log is stopped after a specific period of time has elapsed or at an exact time and date. There are also two options to indicate when the log completes. First, you can specify that when a log file reaches a scheduled termination, another log file begins. This is useful for breaking the log files into smaller, more manageable chunks of data. Second, you could run an executable or batch file. For example, you might want to run a .bat file that includes the *net send* command to alert the administrator that the log is complete.

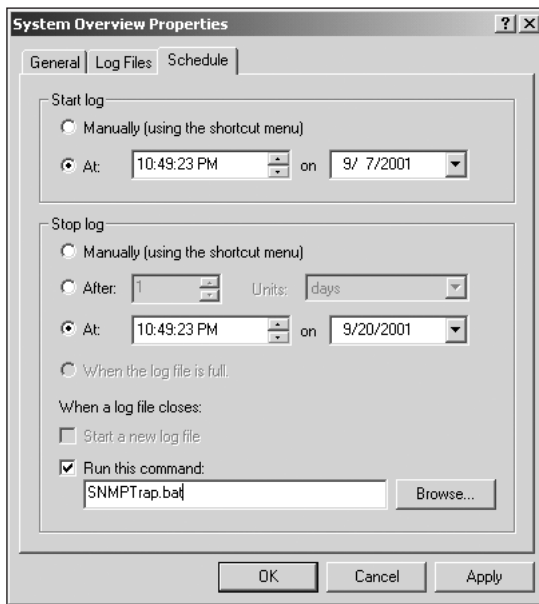


Figure 11-15 Use the Schedule tab to start and stop the log

ESTABLISHING A BASELINE

One of the most important aspects of monitoring performance is establishing a baseline. A **baseline** is established by recording performance data when a server is healthy, or running normally. When problems occur (like the slow logons in the opening scenario), use performance monitoring tools to observe dynamic real-time values. Compare the real-time output with historic performance data to determine the bottleneck in the system. This methodical approach to problem solving will consistently yield faster results than a shotgun approach of random fixes based on experience and luck.

What Is a Bottleneck?

A **bottleneck** refers to the delay in transmission of data through the circuits of a computer's system. When you monitor performance to detect a bottleneck, you are looking for the resource (processor, memory, etc.) that is causing the delay in transmission of data.

Most of us experience bottlenecks every day outside of information technology. Think of a freeway with four lanes heading in the same direction. If traffic is moving at 65 mph on average and a driver moves into the fast (far left) lane and proceeds at 50 mph, this creates a bottleneck. The analogy of a four-lane freeway also relates to servers because there are four basic resources to monitor to create a baseline:

- Processor
- Memory
- Disk subsystem
- Network segment

It is important to note that these are the basic resources, not a comprehensive, detailed picture of a server. We discuss each of these resources in more detail later in this chapter.

When to Create a Baseline

The best time to create a baseline is while the server is experiencing maximum activity. Referencing the logon scenario again, you would create a baseline for the logon server in this network. Record historical performance data during the period of time when the most users are logging on to the system (such as 30–60 minutes after people arrive at work in the morning).

If you were creating a baseline for a database server, the timing might be completely different. Perhaps your company runs the majority of reports on a database server after business hours, between 7 P.M. and 9 P.M. This will be the best time to create the baseline for that database server. You can begin to see why using scheduled performance monitoring (as opposed to manually started monitoring) can be advantageous. In many instances, the optimal monitoring time is not within normal business hours.

What if you are not familiar enough with a server to know the optimal monitoring time? In such instances, use historical performance monitoring to discover the busiest periods of activity. Then create the baseline for that interval.

What to Monitor in a Baseline

Now that you've determined when to monitor performance, the next decision is what resources to monitor. As previously stated, the basics of Processor, Memory, PhysicalDisk and Network Segment are a good place to start. There are exceptions, however, and there is often the need for greater detail.

Returning again to the logon scenario, monitoring the basic resources can provide important information for troubleshooting the slow logon problem. But, based on the specified problem, we can add more objects and create more relevant data. Table 11-3 illustrates possible objects and counters for creating a baseline in the logon scenario for Windows NT or Windows 2000.

Table 11-3 Objects and Counters to Create a Baseline on a Logon Server

Object	Counter
Processor	% Privileged Time
Processor	% User Time
Memory	% Committed Bytes in Use
Server	Logon Total
Server	Logons/sec
Network Segment	% Network Utilization

Note that in addition to the “basic four” resources, we’ve chosen to monitor the Server object, with counters for logon statistics. These additional resources, when included in the baseline, provide specific data about the number of users who normally log on in the recorded time. Even more useful is the number of logons per second. This statistic gives an objective number to use for gauging logon speed. The nature of “fast” or “slow” is very subjective. Note also that the PhysicalDisk object is excluded from this baseline. Disk activity is not a major factor in the logon process.

PUTTING THE TOOLS TO WORK

Let’s continue with the logon scenario to walk through how baseline data and monitoring tools can be used to solve a performance problem. Table 11-4 shows the baseline measurements for the logon server, and Table 11-5 shows the comparative real-time data for the same objects during the Monday morning slowdown.

Table 11-4 Baseline Data for the Logon Server

Object	Counter	Averages (over 30 minutes)
Processor	% Privileged Time	9%
Processor	% User Time	14%
Memory	% Committed Bytes in Use	37%
Server	Logon Total	510 (total over 30 minutes)
Server	Logons/sec	5
Network Segment	% Network Utilization	36%

Table 11-5 Real-Time Data for the Logon Server

Object	Counter	Real-Time Statistics
Processor	% Privileged Time	15%
Processor	% User Time	14%
Memory	% Committed Bytes in Use	39%
Server	Logon Total	1 (one-second snapshot)
Server	Logons/sec	1
Network Segment	% Network Utilization	76%

The first step in interpreting this data is to look for significant changes. In this case, you note the following:

- Processor: % Privileged Time increased by 6%.
- Processor: % User Time is unchanged.
- Memory: % Committed Bytes in Use increased by 2%.
- Server: Logons/sec decreased to 1.
- Network Segment: % Network Utilization increased by 40%.

The most significant changes occurred in the number of logons and network utilization. From this data you can safely say that logons definitely are slow and the bottleneck is the flow of data on the network interface. Solving the problem will require “drilling down” deeper into specifics of network utilization. The value of the baseline is that you have eliminated processor time and memory as possible bottlenecks.



More specifics and possible resolutions of this scenario are addressed later in the chapter.

CAPACITY PLANNING

The baseline measurements for a server can also be used for capacity planning. This is the practice of monitoring resources for the purpose of projecting the effect of increasing or decreasing workload on a server. By measuring the performance of a server under current conditions, we can project how it will perform under another set of conditions. In the current business environment of mergers and acquisitions, capacity planning makes for a smoother IT transition.

Using our logon scenario, the current network has 750 users. Of these, 510 logged on during the performance monitoring that created the baseline in Table 11-4. You learn in a meeting that your company has acquired another company of equal size. Your IT staff has the task of merging IT departments and will be responsible for user logons and

security. You will need to accommodate twice the current number of users on the network. That means 1500 users logging on. Can your server handle the load? Creating baselines for capacity planning will help answer these questions not only for logon servers but also for many network resources.

The sections that follow outline acceptable levels of performance for basic resources (processor, memory, disks, and network utilization) and give solutions to improve performance for each resource. Each of these resources works with the others hand in hand and is capable of influencing the behavior of other resources.

PROCESSOR

Processor time is measured as a percentage of time that the processor is active, executing threads submitted by **processes** (running programs) on the system. (A **thread** is a main component of an application and is the means by which the application accesses memory and processor time). One hundred percent represents constant activity.

Acceptable Processor Performance

A processor running constantly at 100% is overworked and server performance will deteriorate rapidly. Acceptable levels of processor activity extend up to 60–65% on a consistent basis. Levels exceeding 65% during performance monitoring usually indicate that the processor is the bottleneck in the system. However, the specific processor utilization percentage that is acceptable within an organization can vary. For example, perhaps you consider 65% processor utilization to be acceptable for the intranet web server that company employees use. However, for the Internet web transaction server, 65% is way too high, because online purchases will take too long and impatient buyers might cancel transactions.



It is not unusual for the processor to peak or spike higher than 65% for a brief period of time. When new processes are started or when services are starting after rebooting a server, processor levels spiking to 100% are totally acceptable.

A bottleneck is indicated when known applications, processes and/or services push processor levels beyond 65% for an extended period of time and the processor does not return to lower levels until the applications, processes, or services are terminated.

Processor Solutions

The following sections present different approaches to improving processor performance.

Implement SMP

If the processor is the bottleneck, additional CPUs can be added to a server to improve performance and handle increased loads. All major NOSs under discussion in this book support symmetric multiprocessing (SMP). Many 32-bit applications can benefit from

SMP if the code allows **multithreading**, which is the ability to run two or more program threads at once. For example, if a program runs two threads on an SMP system with two processors, each processor can handle a thread simultaneously. With a single processor, the program can still run multiple threads but the processor can only execute a single thread at one time.



Only the simplest programs run a single thread. Most applications (not only on the server but also on most client workstations) run several threads at once.

Add Servers

Sometimes the best solution to a processor bottleneck is to simply add another server, especially when a server is performing multiple tasks that may conflict with each other. For example, a company may be using a single database server to perform sales transactions and provide reports based on those transactions. Transactions and queries for reports may require multiple reads from tables simultaneously. While adding another processor (SMP) may improve performance, a better solution would be to add another server dedicated to running queries to create reports.

Remove Compression

Compression is storing data in a format that requires less space than usual. Simply storing data does not place a greater load on the processor. However, when data is written to the compressed partition or folder, the processor must work harder to calculate the compression algorithms. Removing compression from partitions or folders where data is written frequently can free the processor to perform more critical tasks. The type of data that you choose to compress, if any, is also a factor. Some file types do not compress well, and processor utilization will be wasted on these files. For example, multimedia files such as movie files and JPEG files do not compress well.

Remove Unnecessary Encryption

Encryption uses any of several methods to protect sensitive data from prying eyes. However useful, encryption is processor intensive, and places a greater load on the processor. Just as with compression, the processor performs calculations to encrypt and decrypt data. The operative word in this solution is *unnecessary*. Security is important and when encryption is warranted, the better solution is upgrading or adding additional processors.



Although administrators are usually adept at understanding encryption, you should be careful about users implementing encryption. There are several encryption schemes and utilities available. You do not want users to place encrypted data on network resources where server processors must perform the encryption. In addition, some encryption schemes can make data permanently inaccessible.

Implement Clustering

Clustering is a solution to performance issues that benefit from load balancing. Clustering is connecting two or more computers together in such a way that they behave like a single computer. As a solution to slow processor performance, clustering is essentially adding another computer to aggregate performance in addition to providing fault tolerance.

Remove Software RAID (Especially RAID-5)

RAID (as defined in Chapter 5) provides fault tolerance and in some cases can actually improve performance. Software RAID-5, however, can significantly diminish processor performance. As data is written to the hard disk, the processor must calculate the algorithms for the parity bit that creates fault tolerance. This requires considerable processor time and, consequently, other processes may suffer. If the RAID-5 array is primarily for reading data, this is not an issue because parity calculations are not performed during reads. Hardware RAID-5 does not burden the server CPU because the parity calculation occurs on a separate processor designed for RAID functionality.

Move Processor-Intensive Applications or Services

Moving applications or services that overwork the processor is called load balancing. It includes installing an application or service on a second server, and deleting the application or service from the server that is overworked. For example, if one server is functioning as both the DHCP and WINS server, install WINS on another server and delete WINS from the server with DHCP.

You can also keep the application or service on the original server and then install it on a second server to balance the load between the two servers. This is a common practice in web servers. Instead of overloading a single web server, administrators place the same web content on two or more other web servers, and the web servers take turns in servicing client requests.

Verify Proper Operation of Applications and Drivers

When not running normally, applications or bad drivers can cause excessive processor utilization. To detect problems with applications, monitor the individual process of the application. It will also be useful to monitor the number of threads utilized by the application by using the following object/counter combination:

- Object: Process
- Counter: Thread Count

As an example, in Figure 11-16, Windows 2000 Performance Monitor is monitoring the Diskkeeper defragmentation utility running over four threads (numbered 0–4) and utilizing over 60% processor time. In this case, it was acceptable because I deliberately set Diskkeeper to run at a high priority and there were no other pressing tasks to run at the time.

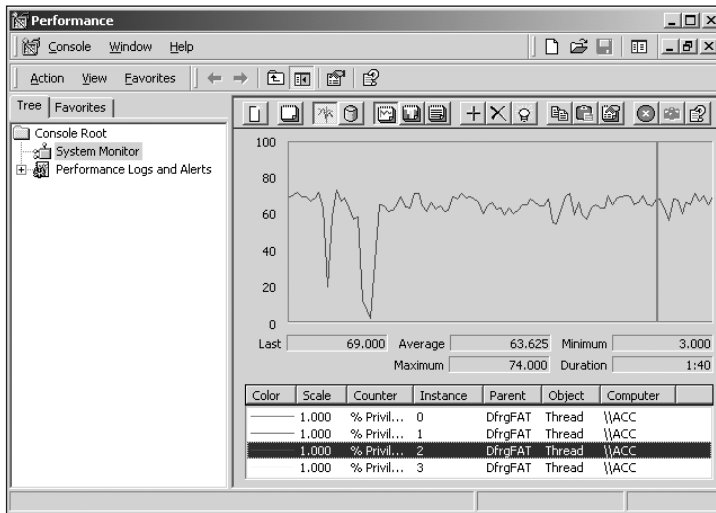


Figure 11-16 Monitor the number of threads in use and the processor utilization they require

A significant change in the number of threads used by an application, compared to a baseline, indicates problems in the application code. Some applications may run multiple instances, which can also increase the load on the processor.

Corrupted device drivers can also demand excessive processor time. An effective measurement for this problem is to monitor the following object/counter combinations:

- Object: Processor
- Counters: % Interrupt Time or Interrupts/sec

Compare the results to a baseline. Significant increases in the number of interrupts indicate problems with hardware devices and/or the drivers. For example, a few years ago, I had a new file server with the best equipment my company could afford at the time. Initially, it performed fine and I largely ignored it except for normal maintenance. One day after clearing dust from inside the server and starting it up, it seemed to take quite a long time to boot. Then, it took a long time to retrieve even the smallest files from the file server. I ran Windows NT Performance Monitor from a different server (so as not to skew the results) to record Interrupts/sec. The interrupts were far above the baseline for this system, and it turned out that the file server's NIC was failing. A failing NIC (and several other types of hardware) will often issue constant interrupts because it is not able to determine that the processor has responded to the interrupt requests. After replacing the NIC, the server returned to its normal level of performance.



Do not be too concerned if Interrupts/sec is over 100 when idle—the system clock accounts for this by sending regular interrupts every 10 milliseconds.

Set Process Priority

In Windows NT and Windows 2000, you can manually set a process or application to run at a specific priority to ensure that it does not dominate processor utilization at the expense of other applications. You can also adjust the process priority to force the processor to favor the process or application over others. Some applications allow you to adjust settings within the application, or you can use Task Manager to configure the application priority:

1. Press Ctrl+Shift+Esc to access Task Manager.
2. On the Process tab, select the application's process.
3. Right-click the process, click Set Priority, and choose a priority from Low to Realtime (see Figure 11-17).

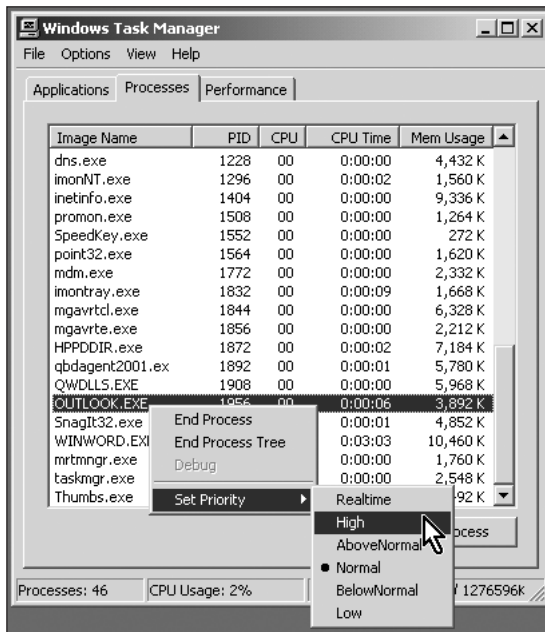


Figure 11-17 Changing the priority level of a process



Use Realtime priority sparingly, if at all, because if the process runs indefinitely, then the operating system, other server functions, or applications can lock or hang due to a lack of processor time.

MEMORY

The following sections present different approaches to improving memory performance. Memory is indirectly one of the most critical performance factors. A low memory condition causes higher disk utilization due to disk swapping. Low memory also affects stability. Some operating system information can never be swapped to hard disk (such as passwords or other security-sensitive data). However, if you are out of memory, there is no place else for it to go, and the server might fail. It is critical to determine what acceptable memory performance is and how to remedy low memory conditions.

Acceptable Memory Performance

Defining acceptable memory performance is extremely subjective. What is tolerable to one organization may be completely unacceptable to another. Most of the time, memory “performance” (that is, speed) is not an issue because it performs in nanoseconds. However, not having enough memory clearly *is* a performance issue, because the NOS must then turn to virtual memory paging on the hard disk—the most common bottleneck component in the system.

The most important counters for memory are:

- Object: Memory
- Counter: % Committed Bytes in Use
- Counter: Page Faults/sec
- Counter: Available Bytes
- Counter: Pages/sec
- Counter: Pool Non-Paged Bytes

Committed Bytes in Use represents a percentage of the total system memory (physical memory plus virtual memory) currently used by processes running on the computer. The first rule of thumb is that this percentage should remain relatively constant. Only slight variations are acceptable. When processes are stopped or started or the number of connected users changes, variation is normal. However, a steady increase of committed bytes, in the absence of additional processes or user load, frequently indicates a **memory leak**. A memory leak occurs when an application opens a thread but does not close it when the application is finished with it. At first, a memory leak will start more paging, which in itself deteriorates performance. Later, as both memory and available swap file space become more scarce, the memory leak can eventually cause a system crash.

Available Bytes is the amount of physical memory available to processes running on the computer. It reflects the last observed value rather than an average.

Pages/sec is a good indicator of excessive paging in a virtual memory system. Paging is a technique to help ensure that the data needed is available as quickly as possible. Recall from Chapter 8 that the page file is designated space on the hard disk used as memory. Each time a page is needed that is not currently in memory, a page fault occurs. When this value exceeds 20 per second on a consistent basis, performance will deteriorate. Excessive page faults are normally due to insufficient memory or memory leaks.

Pool Non-Paged Bytes is the number of bytes in the nonpaged pool, an area of system memory for objects that cannot be written to disk but must remain in physical memory as long as they are allocated. The Registry, for example, cannot be paged to disk.

Memory Solutions

All solutions to memory problems fall under one general category: add more memory. However, it's important to remember that all resources work hand-in-hand; no resource is independent. Increasing or decreasing one resource will *always* have some impact on others. In the case of excessive paging, adding more memory reduces the need for paging and helps to reduce the hard disk bottleneck.

Add Memory

Adding memory can never harm anything except the budget. There are many instances in which adding memory is the best and easiest solution. The best method to determine when additional memory is justified is through performance monitoring. Without performance analysis, simply adding more memory can mask a more serious underlying problem.

Upgrade the Motherboard to Accept More Memory

All motherboards have a limit on the amount and type of memory that can be installed. When this limit is reached, one solution is to upgrade to a motherboard that accommodates more and/or faster memory. Frequently, this is an expensive solution and in some instances not cost effective based on advances in technology. For example, a motherboard that has an older Pentium processor may be limited to 512 MB of RAM. Even if an upgraded motherboard accommodated 2 GB of RAM, significant performance improvements may not be achieved until the processor is also upgraded. Generally, upgrading the motherboard really means replacing the server. The original server becomes spare parts or is used in a less demanding role.

Increase or Optimize Swap File Size

Swap file space goes by different names but is essentially space designated on a hard disk to act as memory. If all physical memory is used and there is not enough swap space, the system will report out-of-memory errors. One solution is to increase the swap file space, but you should realize that this doesn't really improve performance, since the system is still making slower disk accesses instead of rapid memory accesses. Increasing swap file space only prevents more out-of-memory errors.

Depending on the current space dedicated to the swap file, increasing its size may be only a temporary solution. If the swap space or paging file is increased to meet memory needs, performance monitoring will also reveal a corresponding increase in the number of page faults.

Windows-based servers build a page file equal to the amount of RAM by default during installation. NT 4.0 Workstation and Windows 2000 Professional default to 1.5 times physical memory because it is assumed that there is less physical memory in client workstations. The swap space or paging file functions best at a size of 300–500 MB. A previous rule of thumb was 1.5 to 2 times the size of physical RAM, but with physical memory reaching into gigabytes on servers, this number is no longer reasonable. Too much space allotted to a swap file leads to fragmentation. (Fragmentation is discussed later in the chapter.)

You can improve overall system performance regarding swapping to the hard disk by placing the swap file in an optimum location. By default, most NOSs place the swap file on the same hard disk as the operating system itself (often referred to as the system disk). Because the system disk is often quite active running normal operating system services, the swap file must compete for disk access. If available, it is better to place the swap file on a separate physical disk that is less active (another partition on the same disk provides no benefit). Better yet, you can split the swap file between multiple disks (for example, in RAID-0 striping), so that multiple disks can service swap-file activity at once.



If you are trying to determine which of two disks would be best to store the swap file, and all other factors are equal, consider the number of heads the hard disks have. The drive with more heads will perform better.

If the swap file has the ability to grow, as is the case for Windows operating systems, it is better to set a fixed size for the swap file that is large enough to service present and future needs. This helps to prevent fragmentation as the swap file adjusts in size.



As a security precaution, you might consider clearing the swap file when the system reboots. If sensitive information is paged to the swap file and someone is able to gain physical access to it, they might be able to retrieve information from it. This possibility is extremely remote for a number of reasons; however, some highly secure organizations require it. Note that clearing the swap file will cause slower shutdown and startup times while the system clears and re-creates the swap file.

Use Faster Memory

As discussed in Chapter 6, there are different kinds of RAM, some faster than others. SDRAM has almost entirely replaced EDO DRAM and is about twice as fast. SDRAM is capable of synchronizing with the CPU bus and reaching clock speeds of 133 MHz. RDRAM and DDR SDRAM (discussed in Chapter 3) appear to be the next generation of high-performance memory, each capable of more than 1 GBps of data throughput.

Choosing faster memory usually requires a faster motherboard unless, for example, you have 100 MHz SDRAM installed on a 133 MHz bus. In that case, upgrading to 133 MHz SDRAM will take advantage of the faster bus speed.

Distribute Memory-Intensive Applications or Services

As discussed in respect to processors, the best solution to memory problems can be load balancing, because you utilize the hardware resources of another server to alleviate the server load. Thorough performance monitoring and analysis will tell you whether this is the best approach. When monitoring applications and services, pay close attention to which ones are processor intensive and which are memory intensive. A server providing basic file/print services will be memory intensive and probably not place significant demand on the processor. In contrast, a database server providing report functions and servicing multiple queries will be very processor intensive. Familiarity with the relative needs of applications and services in your network will assist you in making the most efficient distribution of resources.

Check for Memory Leaks

Memory leaks were discussed earlier in this chapter in relation to the % Committed Bytes in Use counter. This is perhaps the best indicator of a memory leak. Committed bytes should remain relatively constant. If they continue to increase gradually over time, yet no additional processes are introduced to the server, this is a strong indicator of a memory leak.

The best long-term solution to a memory leak is to contact the vendor so that it can make alterations to the code to stop the leak. Usually, the fix is an update that you can download. The short-term solution is to terminate the application, reboot the server, and restart the application. This forces the application to free memory no longer being used.

HARD DISK

As always, the hard disk seems to be the slowest performing of all the server components. Even with SCSI-3, Fibre Channel, and rotation speeds upward of 15,000 rpm, hard disks cannot begin to compete with the speed of the processor and memory. However, you can still arrive at an acceptable level of performance given the physical limitations of hard disks.

Acceptable Hard Disk Performance

Exact thresholds for determining an acceptable speed for the transfer of data from hard disks or any storage devices are even more subjective than memory or processor performance. The key is to obtain baseline numbers on current performance regardless of

whether it is perceived to be slow or fast. To determine whether the hard disk is able to reasonably keep up with I/O requests, use the following object and counters:

- Object: PhysicalDisk
- Counter: % Disk Time
- Counter: Current Disk Queue Length
- Counter: Avg. Disk Bytes/Transfer

The % Disk Time counter represents the amount of time that the disk services read or write requests. You generally want to see less than 50% for this counter. Current Disk Queue Length represents the number of outstanding I/O requests waiting for the hard disk to become available. If the hard disk is overly taxed, then there will be several outstanding requests. You generally want to see no more than two requests queued. This counter is an instantaneous view; if you want to check an average over time, PhysicalDisk counters such as Avg. Disk Bytes/Transfer are also available.

Hard Disk Solutions

If the time comes when hard disk performance is deemed to be unacceptable, you can implement solutions such as the ones offered below.

Add or Replace Hard Disks

Hardware or software RAID arrays can significantly increase disk performance and provide fault tolerance. Hardware RAID is superior to software RAID, but it is also more expensive. While arguments abound concerning whether software RAID provides true fault tolerance, this is not the forum for that discussion: We're concerned with performance.

Both hardware and software RAID arrays increase performance by striping data across multiple disks. Because multiple drive heads are working simultaneously to write and/or read data, transfer speeds will be faster than non-RAID disks. For example, you have a software RAID-5 array consisting of three hard disks, and performance is unacceptably slow. By adding another disk, you aggregate total performance across four hard disks instead of three (assuming you are using SCSI, not IDE). The only potential problem might be processor utilization for the parity calculation, in which case you might also need to add a processor, upgrade the existing one, or switch to hardware RAID.



Software RAID-5 will increase performance on disk reads. Performance suffers on disk writes, however, due to processor-intensive parity calculations.

Defragment Disks

Fragmentation on hard disks occurs through the normal processes of creating, moving, copying, and deleting files. The result, over time, is that single files are spread out in pieces across the disk. If the condition persists, disk transfer rates deteriorate because the drive head must search around and across multiple sectors to read a single file. Comparing real-time and baseline disk activity can provide evidence of fragmentation.

On a Windows NT/2000 server, use the following object and counter to find evidence of fragmentation:

- Object: PhysicalDisk
- Counter: Disk Read Time

Defragmentation relocates fragmented files back into a contiguous layout. Running a defragmentation utility such as Executive Software's Diskkeeper (see Figure 11-18) on a regular schedule will yield an appreciable increase in performance. Obviously, defragmentation is highly disk intensive, so you should run it only when disk utilization is at its lowest. You can set defragmentation to start on a schedule, or configure defragmentation to start automatically when the hard disk reaches a certain point of fragmentation. In the enterprise, you will want to use Diskkeeper's capability to remotely defragment other servers and workstations.

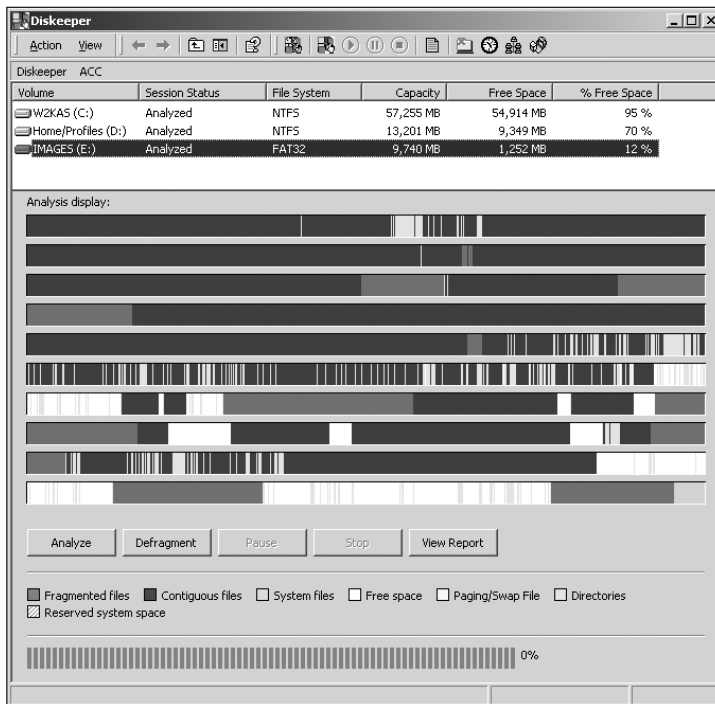


Figure 11-18 Diskkeeper defragments the hard disk



There are not many defragmentation utilities for servers. The most common are Diskeeper (www.executivesoftware.com) and Raxco's PerfectDisk 2000 (www.raxco.com).

Add Faster or Additional Controllers

Adding controllers can solve performance problems. It is very similar to adding another lane to a highway. More controllers accommodate more disks. More disks can balance the workload from multiple users. More controllers and/or disks also increase options for deploying RAID arrays.

The most dramatic and fundamental improvement to the disk subsystem is to upgrade from IDE disks to SCSI disks. Traditionally SCSI controllers or interfaces have supported faster data transfer rates than IDE. The gap has narrowed with the introduction of EIDE, ATA-5, and the upcoming ATA-6. (Recall from Chapter 5 that SCSI-3 supports data transfer rates up to 320 MBps, while ATA-5 and ATA-6 support data transfer rates up to 100 MBps.)

Distribute Files

Distributing files works to solve disk performance problems in a way similar to load balancing. Frequently accessed files are distributed over multiple servers instead of residing on a single server. All network operating systems have some form of distributed files. In the Microsoft environment, it is called the Distributed File System (Dfs). Distributing files has the following advantages:

- There is a single access point for users. In Microsoft Dfs, for example, user computers map to a single file share point and still access files on multiple servers. The share point is the Dfs server, which redirects the requests to the appropriate servers. The process is transparent to users and security is maintained no differently than files accessed normally.
- Distributed files can be a cost-effective performance alternative to adding more servers or upgrading processor, memory, and/or disk resources.

Archive Files to Long-Term Backup Media

As hard disks exceed 75–80% of capacity, performance starts to deteriorate. When large portions of data on a disk must be maintained but not frequently accessed, archiving files to long-term storage can both reduce the risk of running out of disk space and improve performance. Offline storage is available from many hardware and software vendors, but the main idea is that when a given file has not been accessed for specific period of time, the file is automatically moved to offline storage, such as an optical drive or tape. Users can still access the data, but it arrives more slowly as it is retrieved from the offline storage media. Windows 2000 Server integrates this capability into the operating system.

Check for Disk Errors

S.M.A.R.T. is an acronym for Self-Monitoring, Analysis and Reporting Technology. It is an open standard for developing disk drives and software systems that automatically monitor the health of the drive and report potential problems. Potentially, this enables proactive solutions to disk errors before actual disk failure. To use S.M.A.R.T., you load software that is able to query and accept messages from the S.M.A.R.T. hard disk. The software is often provided by the disk manufacturer and included with the hard disk or host adapter. Figure 11-19 shows a hard disk monitoring utility included with Promise Technologies' FastTrak 100 IDE host adapter.

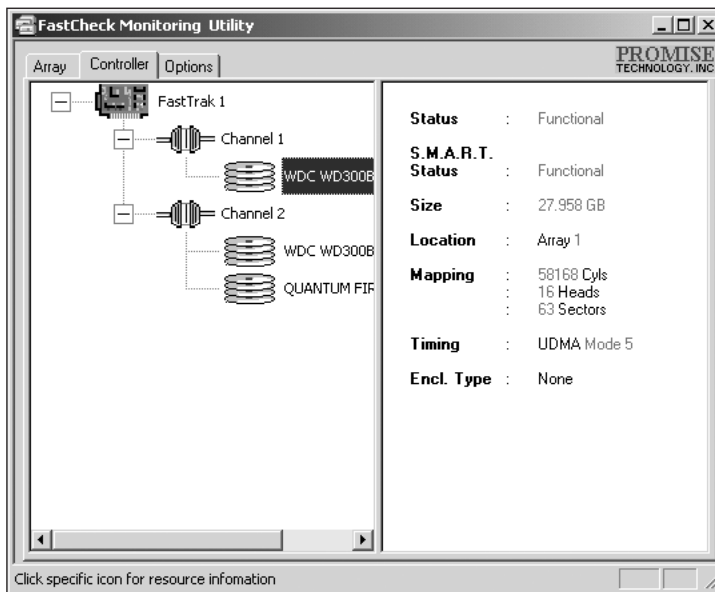


Figure 11-19 This hard disk monitoring utility monitors the health of a hard drive using S.M.A.R.T. reporting

NETWORK

Network performance on a server is contingent on the actual network interface card(s) installed and the components connecting the server to the network, such as cabling switches and/or hubs. Performance Monitor can only measure the traffic on the NICs local to the server.

Acceptable Network Performance

Network utilization is one of the most important network statistics. Most monitoring and reporting tools provide network utilization values as their primary reporting variable. Percentages of up to about 30% network utilization are acceptable. Collision networks (Ethernet) that exceed 30–50% utilization need to be monitored closely to prevent a larger increase of traffic that may cause network delays or low throughput. Server network utilization measures traffic on a specific NIC, and segment utilization measures all traffic on a given segment. Network and server traffic are monitored and analyzed separately, but the acceptable values are the same. Overall network utilization and server network utilization usually affect each other.

To measure the server's ability to send/receive data and handle network requests, monitor the following objects and counters:

- Object: Network Segment
- Counter: % Network Utilization
- Object: Network Interface
- Counter: Output Queue Length

The % Network Utilization counter represents a percentage of network bandwidth in use on the network segment. Each NIC will be an instance of the segment. Output Queue Length measures the number of packets waiting to put out on the network. Values of 1 or 2 are acceptable, but anything higher means your NIC cannot keep up with requests. Multihoming (discussed in the next section) can be a workable solution in this instance.



The Network Monitor Agent service must be installed on Windows NT 4.0/2000 for the Network Segment object to be available in Performance Monitor.

Network Solutions

If you find that network utilization is too high, consider the following solutions.

Multihoming

A multihomed server has two or more NICs installed. Each NIC has a unique IP address. These addresses can be on the same subnet or segmented in two or more subnets. Multiple network cards can notably improve throughput to the server provided network bandwidth is not oversubscribed. More throughput can provide users faster access to data.

There are disadvantages to multihoming some servers because of name resolution issues. A server with two or more NICs will have multiple IP addresses but only one name. In the Microsoft environment, multihomed servers must be manually designated in the WINS

server to guarantee accurate name resolution. DNS names already require manual entry unless you are using dynamic DNS (DDNS), available with Windows 2000 or NetWare. In that case, hosts can automatically register host names to both IP addresses.

Port or Link Aggregation

Introduced in Chapter 7, port aggregation is a software solution to server bandwidth bottlenecks. Port aggregation can double bandwidth to a single segment by combining throughput of two network cards on a single segment. For example, two 10/100 Mbps NICs can be pooled by the software to provide 200 Mbps throughput. Aggregation software also allows a server with multiple NICs to be recognized by a single IP and MAC address. Single IP and MAC addresses overcome the naming resolution problems that can be encountered by standard multihomed servers.

Link aggregation is a hardware and software solution similar to port aggregation. With link aggregation, there is one NIC with multiple ports to provide the additional bandwidth. Link aggregation also requires that only one driver be installed for the network card. This is still a relatively proprietary solution (Intel) and must have compatible switches to deliver maximum performance.



Multihoming, port, and link aggregation only increase the bandwidth to and from the server, but do nothing to increase the overall bandwidth of the network segment.

Improve Network Equipment

Over the past decade, the lower cost and availability of networking equipment has created enormous growth in IT for small and medium-sized companies. One of the key pieces of equipment is the hub (introduced in Chapter 7). Recall that the hub enables multiple nodes to be linked together on a single bus and communicate data to a common destination. Switches can replace hubs to improve network throughput and performance.

A hub with 16 ports accommodates 16 nodes. Using the analogy of highway traffic, think of each of those ports as a lane of traffic. The exit to the bus from the hub is a single lane. Consequently, a hub forces multiple lanes of traffic (in this example, 16) to a single lane. This results in multiple collisions while contesting for the single lane.

A switch with 16 ports also accommodates 16 nodes. However, a switch maintains each lane of traffic to the exit point on the bus. There are no collisions as the lanes are merged. In Ethernet topology, reduced collisions dramatically improve throughput.



Some networks combine switch and hub technology. Hubs, rather than individual nodes, are connected to ports on a switch. While there is no absolute right or wrong to network design, without careful analysis of traffic patterns, arbitrarily connecting hubs to switches can erode the additional throughput provided by the switch.

Upgrade NICs

Ethernet technology as originally developed was capable of transferring data at speeds up to 10 Mbps. Accordingly, the components (NICs, hubs, bridges, routers) developed for networks provided the same throughput speed. In the mid-1990s, Fast Ethernet (100 Mbps) became more affordable and more prevalent in networks.

Upgrading a network card from 10 Mbps to 100 Mbps can boost server throughput and performance. Also consider utilizing full-duplex NICs instead of half-duplex NICs if you are using hubs instead of switches. Recall that full-duplex uses an additional pair of wires and removes collision detection to double potential throughput. As with any other network equipment upgrade, it is only effective if the network bandwidth is not oversubscribed.

Place the Server on the Other Side of a Network Bottleneck

Simple server placement can sometimes eliminate network bottlenecks. You should perform network traffic analysis to identify a bottleneck, but some solutions are simple and logical.

For example, 12 engineers are working on a project and need to share data. Each engineer has a workstation that connects to the network through a hub to the backbone. The engineers have access to a dedicated server named ENGSRV1. This server is located and maintained in a server room, one among many in racks of servers. The engineers complain about slow response time when they save and access data on the server. System administrators analyze server performance and determine the server is handling requests at an acceptable rate—so the problem does not appear to be in the hardware performance on ENGSRV1. The likely conclusion is a network bottleneck between the engineers and the server.

A possible solution to this bottleneck is to swap the hub where the engineers are connected for a 16-port switch. You can connect the server (ENGSRV1) to the same switch with the engineer workstations and bypass the network bottleneck. This is a very simplified solution but introduces the logic to apply when monitoring and analyzing network performance.

CHAPTER SUMMARY

- ❑ Performance monitoring concepts are consistent across all platforms. The specific tools and functions vary according to operating system.
- ❑ System Performance Monitor/2 (SPM/2) is designed to analyze hardware and software in the OS/2 environment.
- ❑ UNIX/Linux text-based tools such as *vmstat*, *ps*, *df*, and *top* are used to monitor performance on Linux systems. You can also use any of several GUI tools, including the GNOME System Monitor, the Stripchart Plotter, and third-party tools such as Unicenter TNG from Computer Associates.
- ❑ NetWare uses the Traffic Manager tool to monitor network traffic. Traffic Manager works with a Windows NT computer within the Performance Monitor tool. To monitor main system resources such as memory, CPU, and hard disk activity, use

the text-based NetWare Monitor utility from the command console. The Java-based ConsoleOne interface also includes a NetWare monitoring utility. To monitor web and FTP servers, use the Novell Internet Caching System (ICS).

- ❑ Performance monitoring on the Windows NT operating system uses a GUI tool named Performance Monitor, and the default view is Chart View.
- ❑ Performance monitoring in Windows 2000 uses the Microsoft Management Console (MMC) graphical interface to the System Monitor, but the objects, counters, and instances are nearly identical to those in the Windows NT 4.0 Performance Monitor tool.
- ❑ The Windows NT/2000 Performance Monitor uses objects, instances, and counters to measure performance on local servers or remote systems. Performance monitoring tools provide both real-time monitoring capability and logging facilities.
- ❑ A baseline is established by recording performance data when a server is healthy, or running normally. The best time to create a baseline is while the server is experiencing maximum activity.
- ❑ When you monitor performance to detect a bottleneck, you are looking for the resource (processor, memory, etc.) that is causing the delay in the transmission of data.
- ❑ Baseline information for a server can also be used for capacity planning.
- ❑ Within a server there are limited resources that can affect the performance of a given system. Each of the resources work hand-in-hand and are capable of influencing the behavior of one another.
- ❑ Processor, Memory, PhysicalDisk, and Network Segment are the basic resources to track in performance monitoring.
- ❑ Processor utilization levels exceeding 65% on a consistent basis during performance monitoring usually indicate that the processor is the bottleneck in the system.
- ❑ SMP can provide improved performance by making multiple CPUs available to complete individual processes simultaneously.
- ❑ Writing data to compressed folders and using unnecessary encryption places an extra load on the CPU.
- ❑ Software RAID-5 can significantly diminish processor performance because the processor must spend resources to calculate the parity bit when writing data.
- ❑ Monitoring swap space is an important aspect of memory management to avoid out-of-memory errors. You can optimize the swap space by spreading it across multiple disks or a RAID array. Avoid placing the swap space on the system disk.
- ❑ Page Faults/sec is a good indicator of excessive paging on a Microsoft server.
- ❑ The best method to determine when additional memory is justified is thorough performance monitoring. The ability to upgrade to more or faster RAM is dependent on the motherboard.

- An undetected memory leak can not only cause performance deterioration, but a system crash as well.
- Hardware or software RAID arrays can significantly increase disk performance and provide fault tolerance. Software RAID-5 will increase performance on disk reads. Performance suffers on disk writes, however, due to processor-intensive parity calculations.
- The most dramatic and fundamental improvement to the disk subsystem is to upgrade from IDE disks to SCSI disks.
- Distributed file systems enable users to map file resources to a single file share point and transparently access resources from many physical locations.
- Overall network utilization and server network utilization usually affect each other.
- Multiple network cards can notably improve throughput to the server.
- Port or link aggregation can double bandwidth to a single segment by combining throughput of two or more network cards on a single segment.
- Upgrading a network card from 10 Mbps to 100 Mbps can boost server throughput and performance.
- Switches can replace hubs to improve network throughput and performance.
- In Ethernet networks, reducing collisions dramatically improves throughput.

KEY TERMS

baseline — A collection of data that establishes acceptable performance. You compare variances in performance against the baseline to determine if perceived performance issues are real.

bottleneck — One or more system components that hinder the performance of the rest of the system. Other system components must wait for the bottleneck item to complete its task before resuming activity.

compression — Data formatted to use less storage space than unformatted data.

counter — In Windows NT and 2000 Performance Monitor, a subset of an object that measures a particular aspect of that object.

drivers — One or more files loaded into the operating system to control a hardware device.

instances — In Windows NT and 2000 Performance Monitor, a subset of object counters that distinguishes like objects from one another. For example, instances would apply to multiple processors, hard disks, or NICs.

memory leak — A program that uses system memory but does not release it when finished. A memory leak consumes memory over time, and causes performance problems because more hard disk virtual memory is required. Eventually, memory leaks can cause a system to return out-of-memory messages or crash.

multithreading — Two or more simultaneously running program threads.

Multithreading is useful for improving performance. Multithreading requires an operating system that can support this, and programmers must be careful to write applications so that threads do not interfere with one another.

objects — In Windows NT/2000 Performance Monitor, resources such as Processor, Memory, PhysicalDisk, and Network Segment.

process — A running program.

System Performance Monitor/2 (SPM/2) — Tool used to measure performance statistics in the OS/2 operating system environment.

threads — Program units of execution that can run separately from other threads. A thread is also the means by which an application accesses memory and processor time.

REVIEW QUESTIONS

1. Which of the following are components of OS/2 performance monitoring?
 - a. Data Collection Facility
 - b. System Monitor
 - c. OS/390
 - d. SPM/2 Monitor
2. The _____ UNIX tool shows resources currently running that consume the most memory.
 - a. *vmstat*
 - b. Logging Facility
 - c. *top*
 - d. Committed Bytes
3. The Windows NT Performance Monitor relies on which two elements to measure performance?
 - a. objects
 - b. cache
 - c. services
 - d. counters
4. The counter Interrupts/sec is associated with which object in NT Performance Monitor?
 - a. Memory
 - b. PhysicalDisk
 - c. Processor
 - d. LogicalDisk

5. Windows 2000 real-time performance is observed with the:
 - a. vmstat utility
 - b. System Monitor
 - c. MMC
 - d. performance logs
6. A test used to compare performance of hardware/software on servers is called a _____.
 - a. bottleneck
 - b. network segment
 - c. baseline
 - d. page fault
7. One or more system components that hinder the performance of the rest of the system is known as a:
 - a. bottleneck
 - b. baseline
 - c. multihoming
 - d. port aggregation
8. To get the most effective comparisons, the best time to create a baseline is:
 - a. between 12:00 A.M. and 6:00 A.M.
 - b. during times of minimal activity
 - c. immediately after rebooting
 - d. during periods of maximum activity
9. Predicting server performance using a current baseline and future conditions is called:
 - a. network planning
 - b. capacity planning
 - c. server planning
 - d. performance planning
10. When data is written to a compressed partition or folder, the processor must:
 - a. remove compression before writing to the disk
 - b. hold the data permanently in memory
 - c. use multiple controllers
 - d. work harder to compress the data before it is written

11. Data encryption affects performance because:
 - a. more memory is required to hold the private key
 - b. more hard disk space is required to store the encryption bits
 - c. additional protocols are necessary to transmit encrypted data over the network
 - d. the processor must perform calculations to encrypt and decrypt the data
12. PC133 SDRAM is capable of synchronizing with the _____ and reaching clock speeds of _____.
 - a. CPU bus/600 MHz
 - b. page file/133 MHz
 - c. CPU bus/133 MHz
 - d. serial port/600 MHz
13. A _____ is a bug in an application or program that prevents it from freeing up memory that it no longer needs.
 - a. memory leak
 - b. page fault
 - c. SCSI
 - d. cluster
14. Software RAID-5 improves performance for _____ operations.
 - a. write
 - b. delete
 - c. read
 - d. copy
15. When files exist in noncontiguous pieces on a hard disk, the condition is known as _____.
 - a. disk performance
 - b. defrag
 - c. disk fragmentation
 - d. IDE fault tolerance
16. SCSI-3 supports data transfer rates up to _____.
 - a. 320 MBps
 - b. 40 Mbps
 - c. 133 MHz
 - d. 600 MHz

17. The server named Infinity is approaching 80% disk capacity. There is no budget to increase disk space at this time. Four other servers are available and can reasonably increase file capacity 10–15%. What is a possible solution?
 - a. add more memory
 - b. upgrade to SCSI
 - c. implement a distributed file system
 - d. rename the server Finite
18. After monitoring a heavily used file server for two hours, the network analysis shows the Output Queue Length averaged a value of 5 and never fell below 3. What is a possible solution?
 - a. add more memory
 - b. upgrade the processor
 - c. upgrade the motherboard
 - d. multihoming
19. Five illustrators work in a remote office across the city from the main office. They each use Windows 2000 Professional workstations. Currently, each illustrator needs to access storyboards on a server in the main office. Only one person accesses the storyboard files from the main office. The illustrators consistently complain about slow access to the server. After monitoring the server, you find it is performing within acceptable parameters. What is a possible solution?
 - a. SCSI controller for the server
 - b. upgrade to faster memory
 - c. move the server to the remote location
 - d. print all files and hire a courier
20. _____ can replace _____ to improve network throughput and performance.
 - a. IDE/SCSI
 - b. hubs/routers
 - c. switches/controllers
 - d. switches/hubs

HANDS-ON PROJECTS



All projects in this chapter are designed for Windows 2000 Server or Professional.



Project 11-1

In this project, you will monitor the Processor object on a Windows 2000 server.

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Performance**. The MMC opens.
2. Click **System Monitor** in the left pane, if necessary.
3. Click the plus sign (+) in the row of buttons above the chart in the right pane. The Add Counters dialog box opens.
4. Click **Use local computer counters**.
5. From the drop-down list under the Performance object, click **Processor**, if it is not already selected.
6. Click **% Processor Time**, if it is not already selected. Click **Add**.
7. Click **% User Time**. Click **Add**.
8. Click **% Privileged Time**. Click **Add**.
9. Click **% Interrupt Time**. Click **Add**. Click **Close**.
10. Start several applications on the server, such as Paint, Word, and Pinball if available. If possible, start a utility that constantly accesses the processor (for example, a 3D screen saver) and preview it. Notice that the counters increase when the screen saver is running. (This is a very good reason not to run fancy screen savers on a server.)
11. Click the first counter, **% Processor Time**. Press **Ctrl+H**. Notice that the processor line in the chart turns white, making it easier to identify when multiple counters are running at once.
12. Click each of the other counters. Note that each chart line turns white as the counter is highlighted.
13. Click **% User Time**. Press the **Del** key to remove the counter. Repeat for % Privileged Time and % Interrupt Time. The % Processor Time counter should remain.
14. Leave System Monitor open for the next project.



Project 11-2

In this project, you will monitor the basic resource objects as defined in this chapter.



The Network Monitor Agent service must be installed to initiate the Network Segment object (Step 7). If necessary, add this using the Add/Remove Programs Control Panel item. Your instructor can help you add this.

1. In System Monitor, click the **(+)** plus sign in the row of buttons above the chart in the right pane. The Add Counters dialog box opens.
2. Click **Use local computer counters**.
3. Click the **Performance object** list box and then click **Memory**.
4. From the counters list, select **% Committed Bytes In Use**. Click **Add**.
5. Click the **Performance object** list box and select **PhysicalDisk**.
6. From the counter list, click **% Disk Time**, if necessary. Click **_Total** from the instance list, if necessary, and then click **Add**.
7. Click the **Performance object** list box and select **Network Segment**.
8. From the counter list, click **% Network Utilization**. Click on the NIC for your subnet in the Instance list. Click **Add**.
9. Click **Close**.
10. Start several applications on the server, such as Paint, Word, and Pinball if available. If possible, start a utility that constantly accesses the processor (for example, a 3D screen saver) and preview it. Notice that the counters increase when the screen saver is running.
11. Click the first counter, **% Processor Time**. Press **Ctrl+H**. Notice that the processor line in the chart turns white, making it easier to identify when multiple counters are running at once.
12. Click each of the other counters. Note that each chart line turns white as the counter is highlighted.
13. Delete all counters and leave the MMC open for Project 11-3.



Project 11-3

In this project, you will configure a performance log using the basic resource objects.

1. Start in the Performance MMC window that is still open from Project 11-2.
2. Click **Performance Logs and Alerts**.
3. In the right pane, right-click **Counter Logs** and click **New Log Settings**.
4. Type **Test Log** in the Name text box, and click OK. The Test Log dialog box opens.
5. Click **Add**. The Select Counters dialog box opens.

6. Click **Use local computer counters**.
7. Click **Processor** from the Performance Object list box. Click % **Processor Time**, and then click **Add**.
8. Click **Memory** from the Performance Object list box. Click % **Committed Bytes In Use**, and then click **Add**.
9. Select **PhysicalDisk** from the Performance Object list box. Click % **Disk Time** from the Counters list. Click **_Total** from the Instances list. Click **Add**.
10. Click **Network Segment** from the Performance Object list box. Click % **Network Utilization** from the Counters list, and click on the NIC for your subnet from the Instance list. Click **Add**, and then click **Close**.
11. Set the Sample Data Interval to 5 seconds.
12. Click the **Log Files** tab. Change the End file names with option to **yyyymmdd**.
13. Click the **Schedule** tab. In the Start log frame, click **Manually**.
14. In the Stop log frame, click the After option and enter **5 minutes**. Click **OK**.
15. Double-click **Counter Logs** in the right pane. Right-click **Test Log**, and click **Start**.
16. Start several applications on the server, such as Paint, Word, and Pinball if available. If possible, start a utility that constantly accesses the processor (for example, a 3D screen saver) and preview it. Notice that the counters increase when the screen saver is running. (This is a very good reason not to run fancy screen savers on a server.)
17. Wait a minimum of five minutes before starting the next exercise. The Test Log icon will turn red when logging is complete. (It may be necessary to refresh the screen by clicking the **Refresh** button or pressing **F5**.)
18. Leave System Monitor open for the next project.



Project 11-4

In this project, you will view the performance log data from the previous project in the Chart Format.

1. Click **System Monitor** in the left pane. Click the **View Log File Data** button (fourth button from the left).
2. Click **Test_Log** and click **Open**.
3. Click the **(+)** plus sign to Add Counters. The Add Counters dialog box opens.
4. Add all objects and counters available. (Note that only the objects and counters selected for logging are available.) Click **Close**. By default, the data appears in the Chart format, which is useful for graphically viewing performance trends from one point in time to the next.
5. Leave Performance Monitor open for the next project.



Project 11-5

In this project, you will view data in the Histogram and Report formats.

1. With the data from Project 11-4 still displayed in Chart format, click the **View Histogram** button (sixth button from the left). This format is useful for viewing log data at a specific point in time.
2. Click the **View Report** button (seventh button from the left). This format is useful for displaying exact numbers for the specified data.
3. Leave the Performance Monitor open for the next project.



Project 11-6

In this project, you will create a performance alert, send a system message, and observe the results in Event Viewer.

1. In the left-hand pane of the Performance window, click **Performance Logs and Alerts**. Right-click **Alerts**, and then click **New Alert settings**.
2. Name the alert **Processor**. Click **OK**. The Processor dialog box opens.
3. Click **Add**. The Select Counters box opens. Select the **Processor** object.
4. Click the **% Processor Time** counter. Click **Add**. Click **Close**.
5. In the “Alert when the value is:” list box, choose **Over**. Enter **5** in the Limit box.
6. Choose to sample data every 20 seconds.
7. Click the **Action** tab. Accept the default to Log an entry in the application event log.
8. Check the box to send a network message. Enter your computer name.
9. Click the **Schedule** tab. Click the **Start scan Manually** radio button.
10. Choose **Stop scan After 1 minute**. Click **OK**.
11. Right-click the **Processor** alert. Select **Start**. Open several applications to increase processor time. You will begin receiving system messages. Click **OK** to acknowledge the messages. Wait one minute.
12. Click **Start**, point to **Programs**, point to **Administrative Tools**, and click **Event Viewer**.
13. Click the **Application Log**. Note the messages indicating that the processor exceeded the limit set in the alert.
14. Close all open windows.

CASE PROJECTS



1. It's your first week on the job as the administrator. Complaints roll in that logon is slow. When you ask for performance logs on logons, you get blank looks except for one guy who tried once but couldn't remember where the data was saved. You decide to monitor some performance elements yourself, in order to identify what the bottleneck might be.

Based on the data in Table 11-6, what is the likely bottleneck and a possible solution?

Table 11-6 Sample Data for Case Project 11-1 (Values Are Averages Based on One-Minute Monitor Times)

Object	Counter	Value
Processor	% Privileged Time	84%
Processor	% User Time	52%
Memory	% Committed Bytes in Use	44%
Server	Logon Total	63
Server	Logon/sec	1
Network Segment	% Network Utilization	37%

2. Day 2 on the new job and you are downloading files from the primary file/print server. While no one has complained about the slow downloads, the performance is not acceptable to you. You use Performance Monitor to establish a baseline. Based on the data in Table 11-7, identify the most likely bottleneck and suggest a solution.

Table 11-7 Sample Data for Case Project 11-2 (Values Are Averages Based on 30-Minute Performance Log)

Object	Counter	Value
Processor	% Privileged Time	37%
Processor	% User Time	26%
Memory	Page Faults/sec	56
PhysicalDisk	% Disk Time	77%
Network Segment	% Network Utilization	37%

3. The salesman for a new integrated contact management software package is tout-
ing the virtues of his wares to your marketing manager. The salesman leaves
behind an evaluation copy of the server software. The marketing manager wants
to try it out immediately, of course. Like a smart administrator, you install the
software on a test system. The next morning, you record a performance log to
track the impact of the software. Based on the data in Table 11-8, is there a bot-
tleneck? Is there any evidence of potential problems?

Table 11-8 Sample Data for Case Project 11-3 (Values Are Averages Based on
240-Minute Performance Log)

Object	Counter	Value
Processor	% Privileged Time	12%
Processor	% User Time	9%
Memory	% Committed Bytes in Use	22–53%
PhysicalDisk	% Disk Time	17%
Network Segment	% Network Utilization	8%